Claim Amendments

Please amend the Claims as follows:

1-49 (cancelled)

50. (Currently amended) A process for recognizing a digital image using a computer and an ABM algorithm, where ABM is the "Attrasoft Boltzmann Machine, which further consists of the ABM training algorithm and the ABM recognition algorithm, said ABM training algorithm comprises: ~~, said image comprising:~~

Imposing an image to an ABM so the ABM will be trained, whereas the ABM is a specific combination of (a) a fully connected neural network and (b) a Markov chain;

Classifying at least one target image based on the invariant distribution function of the trained ABM wherein the step of Imposing an image to an ABM further comprises:

a) deleting existing ABM connections;
b) creating an input vector, p, based on an input image, x, and its classification, y;
c) breaking the input vector, p, into a number of pieces, p1, p2, p3..., where such breaking could either be logically based on objects/segments or geometrically (such as a division of 10 equal parts);
d) constructing a set of neural state vectors, s1, s2, s3 ... according to p1, p2, p3..., whereas a state vector, s1, has a number of 0's (grounded state) and a number of 1's (excited state); all such vectors together form a configuration space, H(P);
e) computing an initial neural connection from each of p1, p2, p3..., said computation comprising:

> e1) constructing a connection space, H(C), where each neural connection is a point inside this space;
> e2) making the connection space, H(C), from a configuration space, H(C) = (H(P), R), where R is a space of real numbers;
> e3) making an initial connection c1 to be c1 = (p1, 1), or f (p1) = 1, where f(p1) is a connection matrix element;

f) computing the rest of the neural connections from each of the initial connections, c1, c2, c3..., said computation comprising:

> f1) constructing a distance or distances, d (p1, p1'), between an initial neural state, p1, and an arbitrary state, p1'; said distances can be Hausdorff distance, or L1 distance, or L2 distance, or any other distances;
> f2) constructing a function, g(d), which maps a distance between two neural vectors, d, to a number, g(d); said function comprising of any functions as long as it decreases in value when the distance increases, for example, g(d) = 1 / (1 + d) or g(d) = 1 / (1 + d + d * d);
> f3) constructing an arbitrary connection element (p1', g (d (p1, p1'))) from the initial connection element;

f4) applying (p1', g (d (p1, p1'))) for all points in the connection space since the ABM is a fully connected network with all possible ranks.

g) constructing an ABM Markov chain after all of the connections are established.

51. (cancelled)

52. (Currently amended) The process of Claim 50, wherein the step of Classifying at least one target image based on the invariant distribution function of the trained ABM comprises:

a) imposing an image to be classified on an ABM Markov chain;
b) allowing the ABM Markov chain to settle on its invariant distribution, described by a distribution function;
c) classifying the target image based on this invariant distribution function, said distribution comprising of information of classes ~~entered in Claim 36 (b)~~ and weight given by the invariant distribution function directly;
d) presenting the results as a triplet (image, classification, weight), which can be used both in image search and image classification.

53. (Currently amended) A process for recognizing a digital image using a computer and an APN algorithm, where APN is "Attrasoft PolyNet", an extension of the ABM, which further consists of the APN training algorithm and the APN recognition algorithm, said APN training algorithm comprises: ~~A process for recognizing a digital image using a computer, said image comprising:~~

Imposing an image to an APN so the APN will be trained, whereas the APN is a specific combination of (a) a fully connected neural network, (b) a Markov chain, and (c) a mapping function (called the APN function);

Classifying at least one target image based on the invariant distribution function of the trained APN wherein the step of Imposing an image to an APN further comprises:

a) deleting existing APN connections;
b) creating an input vector, p, based on an input image, x, and its classification, y;
c) breaking the input vector, p, into a number of pieces, p1, p2, p3..., where such breaking could either be logically based on objects/segments or geometrically (such as a division of 10 equal parts);
d) constructing a set of neural state vectors, s1, s2, s3 ... according to p1, p2, p3..., whereas a state vector, s1, has a number of 0's (grounded state) and a number of 1's (excited state); all such vectors together form a configuration space, H(P);
e) computing an initial neural connection from each of p1, p2, p3..., said computation comprising:

    e1) constructing a connection space, H(C), where each neural connection is a point inside this space;
    e2) making the connection space, H(C), from a configuration space, H(C) = (H(P), R), where R is a space of real numbers;
    e3) making an initial connection c1 to be c1 = (p1, 1), or f (p1) = 1, where f(p1) is a connection matrix element;

f) computing the rest of the neural connections from each of the initial connections, c1, c2, c3..., said computation comprising:

f1) constructing a distance or distances, d (p1, p1'), between an initial neural state, p1, and an arbitrary state, p1'; said distances can be Hausdorff distance, or L1 distance, or L2 distance, or any other distances;

f2) constructing a function, g(d), which maps a distance between two neural vectors, d, to a number, g(d); said function comprising of any functions as long as it decreases in value when the distance increases, for example, g(d) = 1 / (1 + d) or g(d) = 1 / (1 + d + d * d);

f3) constructing an arbitrary connection element (p1', g (d (p1, p1'))) from the initial connection element;

f4) applying (p1', g (d (p1, p1'))) for all points in the connection space since the APN is a fully connected network with all possible ranks;

g) constructing an APN Markov chain after all of the connections are established;

h) constructing an initial mapping to reflect the contribution of the multi-valued neurons; said mapping comprising of pairs: (neuron position, input vector value);

i) constructing the rest of the mappings based on the initial mapping; said mapping comprising of pairs: (neuron position, input vector value in the initial mapping).

54. (Cancelled)

55. (Currently amended) The process of Claim 53, wherein the step of Classifying at least one target image based on the invariant distribution function of the trained APN comprises:

a) imposing an image to be classified on an APN Markov chain;
b) allowing the APN Markov chain to settle on its invariant distribution, described by a distribution function;
c) classifying the target image based on this invariant distribution function; said distribution comprising of information of classes and weight given by the invariant distribution function directly;
d) constructing intermediate results as a triplet (image, classification, weight);
e) computing a new weight, said computation comprising:
   e1) constructing a distance between two mappings, the mapping associated with the image to be classified and the mapping associated with a the connection created in Claim 54 (h), (i) ;
   e2) constructing a function (called the APN function): weight' = h (weight, d); said function, h, comprising of any functions as long as it decreases in value when the distance, d, increases;
   e3) modifying the old weight by this function;
f) constructing results as a triplet (image, classification, weight'), which can be used both in image search and image classification.

Claim Amendments

Please amend the Claims as follows:

1-49 (cancelled)

50. (Currently amended) A process for recognizing a digital image using a computer and an ABM algorithm, where ABM is the "Attrasoft Boltzmann Machine, which further consists of the ABM training algorithm and the ABM recognition algorithm, said ABM training algorithm comprises: , said image comprising:

Imposing an image to an ABM so the ABM will be trained, whereas the ABM is a specific combination of (a) a fully connected neural network and (b) a Markov chain;

Classifying at least one target image based on the invariant distribution function of the trained ABM wherein the step of Imposing an image to an ABM further comprises:

   a) deleting existing ABM connections;
   b) creating an input vector, p, based on an input image, x, and its classification, y;
   c) breaking the input vector, p, into a number of pieces, p1, p2, p3..., where such breaking could either be logically based on objects/segments or geometrically (such as a division of 10 equal parts);
   d) constructing a set of neural state vectors, s1, s2, s3 ... according to p1, p2, p3..., whereas a state vector, s1, has a number of 0's (grounded state) and a number of 1's (excited state); all such vectors together form a configuration space, H(P);
   e) computing an initial neural connection from each of p1, p2, p3..., said computation comprising:

   > e1) constructing a connection space, H(C), where each neural connection is a point inside this space;
   > e2) making the connection space, H(C), from a configuration space, H(C) = (H(P), R), where R is a space of real numbers;
   > e3) making an initial connection c1 to be c1 = (p1, 1), or f (p1) = 1, where f(p1) is a connection matrix element;

   f) computing the rest of the neural connections from each of the initial connections, c1, c2, c3..., said computation comprising:

   > f1) constructing a distance or distances, d (p1, p1'), between an initial neural state, p1, and an arbitrary state, p1'; said distances can be Hausdorff distance, or L1 distance, or L2 distance, or any other distances;
   > f2) constructing a function, g(d), which maps a distance between two neural vectors, d, to a number, g(d); said function comprising of any functions as long as it decreases in value when the distance increases, for example, g(d) = 1 / (1 + d) or g(d) = 1 / (1 + d + d * d);
   > f3) constructing an arbitrary connection element (p1', g (d (p1, p1'))) from the initial connection element;

f4) applying (p1', g (d (p1, p1'))) for all points in the connection space since the ABM is a fully connected network with all possible ranks.

g) constructing an ABM Markov chain after all of the connections are established.

51. (cancelled)

52. (Currently amended) The process of Claim 50, wherein the step of Classifying at least one target image based on the invariant distribution function of the trained ABM comprises:

a) imposing an image to be classified on an ABM Markov chain;
b) allowing the ABM Markov chain to settle on its invariant distribution, described by a distribution function;
c) classifying the target image based on this invariant distribution function, said distribution comprising of information of classes ~~entered in Claim 36 (b)~~ and weight given by the invariant distribution function directly;
d) presenting the results as a triplet (image, classification, weight), which can be used both in image search and image classification.

53. (Currently amended) A process for recognizing a digital image using a computer and an APN algorithm, where APN is "Attrasoft PolyNet", an extension of the ABM, which further consists of the APN training algorithm and the APN recognition algorithm, said APN training algorithm comprises:
~~A process for recognizing a digital image using a computer, said image comprising:~~

Imposing an image to an APN so the APN will be trained, whereas the APN is a specific combination of (a) a fully connected neural network, (b) a Markov chain, and (c) a mapping function (called the APN function);

Classifying at least one target image based on the invariant distribution function of the trained APN wherein the step of Imposing an image to an APN further comprises:

a) deleting existing APN connections;
b) creating an input vector, p, based on an input image, x, and its classification, y;
c) breaking the input vector, p, into a number of pieces, p1, p2, p3..., where such breaking could either be logically based on objects/segments or geometrically (such as a division of 10 equal parts);
d) constructing a set of neural state vectors, s1, s2, s3 ... according to p1, p2, p3..., whereas a state vector, s1, has a number of 0's (grounded state) and a number of 1's (excited state); all such vectors together form a configuration space, H(P);
e) computing an initial neural connection from each of p1, p2, p3..., said computation comprising:

e1) constructing a connection space, H(C), where each neural connection is a point inside this space;
e2) making the connection space, H(C), from a configuration space, H(C) = (H(P), R), where R is a space of real numbers;
e3) making an initial connection c1 to be c1 = (p1, 1), or f (p1) = 1, where f(p1) is a connection matrix element;

f) computing the rest of the neural connections from each of the initial connections, c1, c2, c3…, said computation comprising:

f1) constructing a distance or distances, d (p1, p1'), between an initial neural state, p1, and an arbitrary state, p1'; said distances can be Hausdorff distance, or L1 distance, or L2 distance, or any other distances;

f2) constructing a function, g(d), which maps a distance between two neural vectors, d, to a number, g(d); said function comprising of any functions as long as it decreases in value when the distance increases, for example, g(d) = 1 / (1 + d) or g(d) = 1 / (1 + d + d * d);

f3) constructing an arbitrary connection element (p1', g (d (p1, p1'))) from the initial connection element;

f4) applying (p1', g (d (p1, p1'))) for all points in the connection space since the APN is a fully connected network with all possible ranks;

g) constructing an APN Markov chain after all of the connections are established;

h) constructing an initial mapping to reflect the contribution of the multi-valued neurons; said mapping comprising of pairs: (neuron position, input vector value);

i) constructing the rest of the mappings based on the initial mapping; said mapping comprising of pairs: (neuron position, input vector value in the initial mapping).

54. (Cancelled)

55. (Currently amended) The process of Claim 53, wherein the step of Classifying at least one target image based on the invariant distribution function of the trained APN comprises:

a) imposing an image to be classified on an APN Markov chain;

b) allowing the APN Markov chain to settle on its invariant distribution, described by a distribution function;

c) classifying the target image based on this invariant distribution function; said distribution comprising of information of classes and weight given by the invariant distribution function directly;

d) constructing intermediate results as a triplet (image, classification, weight);

e) computing a new weight, said computation comprising:

e1) constructing a distance between two mappings, the mapping associated with the image to be classified and the mapping associated with a ~~the~~ connection ~~created in Claim 54 (h), (i)~~ ;

e2) constructing a function (called the APN function): weight' = h (weight, d); said function, h, comprising of any functions as long as it decreases in value when the distance, d, increases;

e3) modifying the old weight by this function;

f) constructing results as a triplet (image, classification, weight'), which can be used both in image search and image classification.